Unstructured Multigrid Through Agglomeration

V. Venkatakrishnan
Computer Sciences Corporation
M.S. T045-1, NAS Applied Research Branch
NASA Ames Research Center, Moffett Field, CA 94035
D. J. Mavriplis
ICASE
M.S. 132C
NASA Langley Research Center, Hampton, VA 23681
M.J. Berger
Courant Institute of Mathematical Sciences
New York University, New York, NY 10012

Abstract

In this work the compressible Euler equations are solved using finite volume techniques on unstructured grids. The spatial discretization employs a central difference approximation augmented by dissipative terms. Temporal discretization is done using a multistage Runge-Kutta scheme. A multigrid technique is used to accelerate convergence to steady state. The coarse grids are derived directly from the given fine grid through agglomeration of the control volumes. This agglomeration is accomplished by using a greedy-type algorithm and is done in such a way that the load, which is proportional to the number of edges, goes down by nearly a factor of 4 when moving from a fine to a coarse grid. The agglomeration algorithm has been implemented and the grids have been tested in a multigrid code. An area-weighted restriction is applied when moving from fine to coarse grids while a trivial injection is used for prolongation. Across a range of geometries and flows, it is shown that the agglomeration multigrid scheme compares very favorably with an unstructured multigrid algorithm that makes use of independent coarse meshes, both in terms of convergence and elapsed times.

1 Introduction

Multigrid techniques have been successfully used in computational aerodynamics for over a decade [1, 2]. The main advantage of the multigrid method when solving steady flows is the enhanced convergence while requiring little additional storage. In addition, multigrid can be used in conjunction with any convergent base scheme, with adequate care exercised in constructing proper restriction and prolongation operators between the grids. Perhaps the biggest advantage of multigrid is the fact that it deals directly with the nonlinear problem without requiring an elaborate linearization and the attendant storage required to store the matrix that arises from the linearization. Thus, multigrid techniques have enabled the practical solution of complex aerodynamic flows using millions of grid points.

The initial efforts in multigrid were directed towards the solution of flows on structured grids where coarse grids can easily be derived from a given fine grid. Typically, this is done by omitting alternate grid lines in each dimension. These ideas have been extended to triangular

grids in two dimensions and to tetrahedral meshes in three dimensions [3, 4, 5, 6]. In previous work by the second author, a sequence of unnested triangular grids of varying coarseness is constructed [3]. Piecewise linear interpolation operators are derived during a preprocessing step by using efficient search procedures. The residuals are restricted to coarse grids in a conservative manner. It has been shown that such a scheme can consistently obtain convergence rates comparable to those obtained with existing structured grid multigrid methods. For complex geometries, especially in three dimensions, however, constructing coarse grids that faithfully represent the complex geometries can become a difficult proposition. Thus, it is often desirable to derive the coarse grids directly from a given fine grid.

The agglomeration multigrid strategy has been investigated by Lallemand et al. [7] and Smith [8]. Lallemand et al. use a base scheme where the variables are stored at the vertices of the triangular mesh, whereas Smith uses a scheme that stores the variables at the centers of triangles. In the present work, a vertex-based scheme is employed. Two dimensional triangular grids contain twice as many cells as vertices (neglecting boundary effects), and three dimensional tetrahedral meshes contain 5 to 6 times more cells than vertices. Thus, on a given grid, a vertex scheme incurs substantially less computational overhead than a cell-based scheme. Increased accuracy can be expected from a cell-based scheme, since this involves the solution of a larger number of unknowns. However, the increase in accuracy does not appear to justify the additional computational overheads, particularly in three dimensions.

The main idea behind the agglomeration strategy of Lallemand et al. [7] is to agglomerate the control volumes for the vertices using heuristics. The centroidal dual, composed of segments of the median of the triangulation, is a collection of the control volumes over which the Euler equations in integral form are solved. On simple geometries, Lallemand et al. were able to show that the agglomerated multigrid technique performed as well as the multigrid technique which makes use of unnested coarse grids. However, the convergence rates, especially for the second order accurate version of the scheme, appeared to degrade somewhat. Furthermore, the validation of such a strategy for more complicated geometries and much finer grids, as well as the incorporation of viscous terms for the Navier-Stokes equations, remains to be demonstrated. The work of Smith [8] constitutes the basis of a commercially available computational fluid dynamics code, and as such has been applied to a number of complex geometries [9]. However, consistently competitive multigrid convergence rates have yet to be demonstrated.

In the present work, the agglomeration multigrid strategy is explored further. The issues involved in a proper agglomeration and the implications for the choice of the restriction and prolongation operators are addressed. Finally, flows over non-simple two-dimensional geometries are solved with the agglomeration multigrid strategy. This approach is compared with the unstructured multigrid algorithm of Mavriplis [3] which makes use of unnested coarse grids. Convergence rates as well as CPU times on a Cray Y-MP/1 are compared using both methods.

2 Governing equations and discretization

The Euler equations in integral form for a control volume Ω with boundary $\partial\Omega$ read

$$\frac{d}{dt}\int_{\Omega} u \, dv + \oint_{\partial\Omega} F(u, n) \, dS = 0. \tag{1}$$

Here u is the solution vector comprised of the conservative variables: density, the two components of momentum, and total energy. The vector $F(u, n)$ represents the inviscid flux vector for a surface with normal vector n. Equation (1) states that the time rate of change of the variables inside the control volume is the negative of the net flux of the variables through the boundaries of the control volume. This net flux through the control volume boundary is termed the *residual*. In the present scheme the variables are stored at the vertices of a triangular mesh. The control volumes are non-overlapping polygons which surround the vertices of the mesh. They form the *dual* of the mesh, which is composed of segments of medians. Associated with each edge of the original mesh is a (segmented) dual edge. The contour integrals in Equation (1) are replaced by discrete path integrals over the edges of the control volume. Figure 1 shows a triangulation for a four-element airfoil and Figure 2 shows the centroidal dual. Each cell in Figure 2 represents a control volume. The path integrals are computed by using the trapezoidal rule. This can be shown to be equivalent to using a piecewise linear finite-element discretization. For dissipative terms, a blend of Laplacian and biharmonic operators is employed, the Laplacian term acting only in the vicinity of shocks. A multi-stage Runge-Kutta scheme is used to advance the solution in time. In addition, local time stepping, enthalpy damping and residual averaging are used to accelerate convergence. The principle behind the multigrid algorithm is that the errors associated with the high frequencies are annihilated by the carefully chosen smoother (the multi-stage Runge-Kutta scheme) while the errors associated with the low frequencies are annihilated on the coarser grids where these frequencies manifest themselves as high frequencies. In previous work [3], as well as in the present work, only the Laplacian dissipative term (with constant coefficient) is used on the coarse grids. Thus the fine grid solution itself is second order accurate, while the solver is only first order accurate on the coarse grids.

3 Details of agglomeration

The agglomeration (referred to also as coarsening) algorithm is a variation on the one used by Lallemand et al. [7] and is given below:

1. Pick a starting vertex on the surface of one of the airfoils.
2. Agglomerate control volumes associated with its neighboring vertices which are not already agglomerated.
3. Define a front as comprised of the exterior faces of the agglomerated control volumes. Place the exposed edges in a queue.
4. Pick the new starting vertex as the unprocessed vertex incident to a new starting edge which is chosen from the following choices given by order of priority:

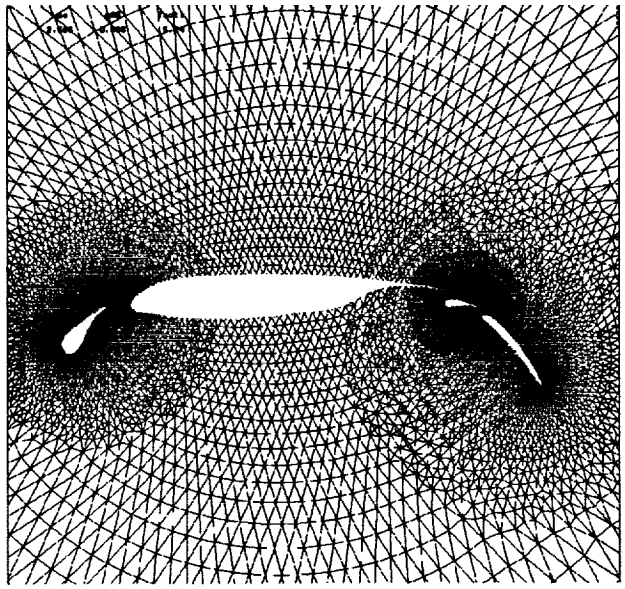

Figure 1: Grid about a four-element airfoil.

- An edge on the front that is on the solid wall.
- An edge on the solid wall.
- An edge on the front that is on the far field boundary.
- An edge on the far field boundary.
- The first edge in the queue.

5. Go to Step 2 until the control volumes for all vertices have been agglomerated.

There are many other ways of choosing the starting vertex in Step 4 of the algorithm, but we have found the above strategy to be the best. The efficiency of the agglomeration technique can be characterized by a histogram of the number of fine grid cells comprising each coarse grid cell. Ideally, each coarse grid cell will be made up of exactly four fine grid cells. The various strategies can be characterized by how close they come to this ideal case. One variation is to pick the starting edge randomly from the edges currently on the front. Figure 3 shows a plot of the number of coarse grid cells as a function of the number of fine grid cells comprising them, with our agglomeration algorithm described above, and with the variation. It is clear that our agglomeration algorithm is superior to the variant. The number of coarse grid cells having exactly one fine cell (singletons) is also much smaller with our algorithm compared to the variant. We have also investigated another variation where the starting vertex in Step 4 is randomly picked from the field and this turns out to be much worse. It is possible to identify the singleton cells and agglomerate them with the neighboring cells, but this has not been done.

The procedure outlined above is applied recursively to create coarser grids. Figure 4 shows an example of the agglomerated coarse grid. The boundaries between the control volumes on the coarse grids are composed of the edges of the fine grid control volumes. We have observed that the number of such edges only goes down by a factor of 2 when going from a fine to a coarse grid. Since the computational load is proportional to the number of edges,

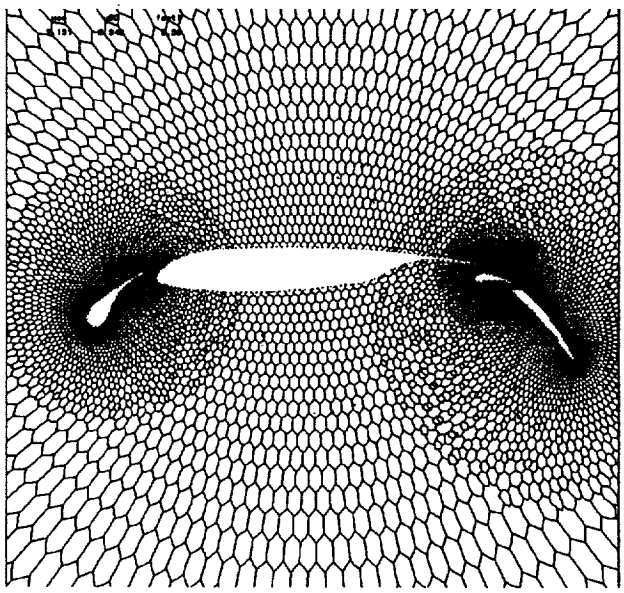

Figure 2: Centroidal dual for the triangulation of Figure 1.

this is unacceptable in the context of multigrid. However, if we recognize that the multiple edges separating two control volumes can be replaced by a single edge connecting the end points, then the number of edges does go down by a factor of 4. Since only a first order discretization is used on the coarse grids, there is no approximation involved in this step. If a flux function that involved the geometry in a nonlinear fashion were used, such as the Roe's approximate Riemann solver, this is still a very good approximation. It may also be seen from Figure 4 that once this approximation is made, the degree of a node in this graph is still 3 i.e., each node in the interior has precisely three edges emanating from it. Thus the agglomerated grid implies a triangulation of the vertices of a dual graph of the coarse grid. Trying to reconstruct the triangulation is not a good idea, since this may result in a graph with intersecting edges (non planar graph), which leads to non-valid triangulations. If a valid triangulation could always be constructed, it would be possible to use the coarse grid triangulation for constructing piecewise linear operators for prolongation and restriction akin to the non-nested multiple grid scheme [3]. In practice, we have often found the implied coarse grid triangulations to be invalid and therefore the coarse grids are only defined in terms of control volumes. This has some important implications for the multigrid algorithm discussed below.

Since the fine grid control volumes comprising a coarse grid control volume are known, the restriction is similar to that used for structured grids. The residuals are simply summed from the fine grid cells and the variables are interpolated in an area-weighted manner. For the prolongation operator, we use a simple injection (a piecewise constant interpolation). This is an unfortunate but unavoidable consequence of using the agglomeration strategy. A piecewise linear prolongation operator implies a triangulation, the avoiding of which is the main motivation for the agglomeration. However, additional smoothing steps may be employed to minimize the adverse impact of the injection. This is achieved by applying an averaging procedure to the injected corrections. In an explicit scheme, solution updates are directly proportional to the computed residuals. Thus, by analogy, for the multigrid scheme, correc-

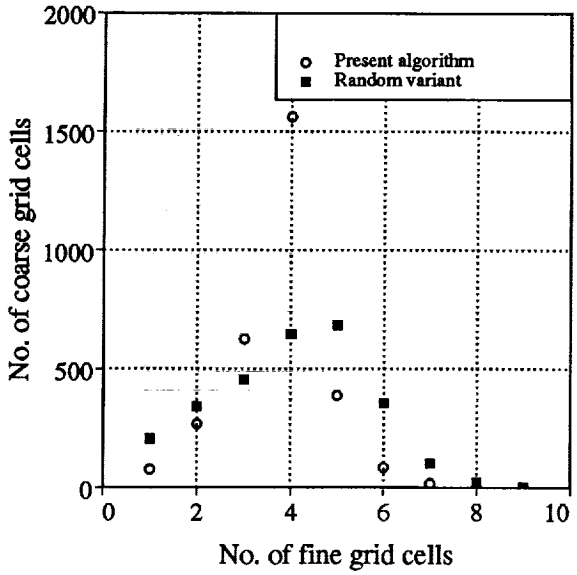

Figure 3: No. of coarse grid cells as a function of the fine grid cells they contain.

tions may be smoothed by a procedure previously developed for implicit residual smoothing [3]. The implicit equations for the smoothed corrections are solved using two iterations of a Jacobi scheme after the prolongation at each grid level.

The agglomeration step is done as a preprocessing operation on a workstation. It is very efficient and employs hashing to combine the multiple fine grid control volume edges separating two coarse grid cells into one edge. The time taken to derive 5 coarse grids on a Silicon Graphics work station model 4D/25 (20 MHz clock) for the grid shown in Figure 1 with 11340 vertices is 83 seconds.

4 Results and discussion

Results are presented for two inviscid flow calculations and the performance of the agglomerated multigrid algorithm is compared with that of the non-nested multiple grid multigrid algorithm of [3]. The first flow considered is flow over an NACA0012 airfoil at a freestream Mach number of 0.8 and angle of attack of 1.25°. The dual to the fine grid having 4224 vertices is shown in Figure 5. The sequence of unnested grids (not shown) for use with the non-nested multigrid algorithm contains 1088, 288 and 80 vertices, respectively. The agglomerated grids are shown in Figure 6. These grids have 1088, 288 and 80 vertices (regions) as well. Figure 7 shows the convergence histories obtained with the non-nested and agglomeration multigrid algorithms. Both the multigrid strategies employ W-cycles. The convergence histories show that the multigrid algorithm slightly outperforms the agglomeration algorithm. The CPU times required for 100 iterations on the Cray Y-MP/1 are 25 and 24 seconds, respectively. Thus the two schemes perform equally well.

The next case considered is flow over a four-element airfoil. The freestream Mach number is 0.2 and the angle of attack is 5°. The fine grid has 11340 vertices and is shown in Figure 1. The coarse grids for use with the non-nested multigrid algorithm (not shown) contain 2942 and 727 vertices. The two agglomerated grids are shown in Figure 8. These grids contain

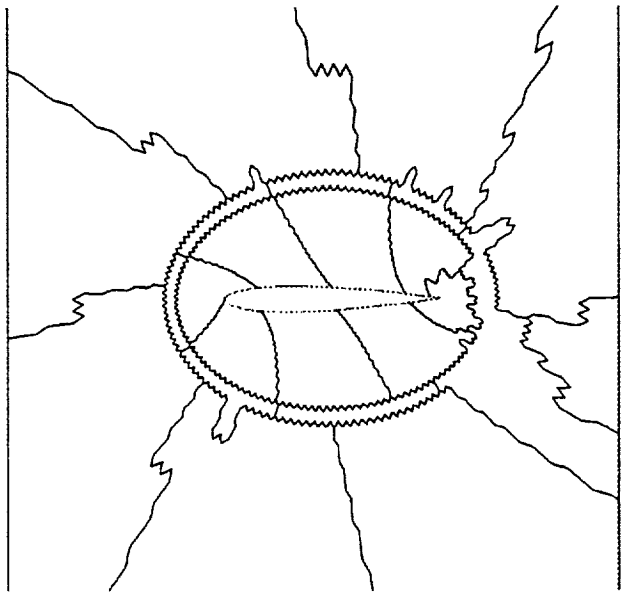

Figure 4: An example of an agglomerated coarse grid.

3027 and 822 vertices (regions), respectively. The convergence histories of the non-nested and agglomeration multigrid algorithms are shown in Figure 9. The convergence histories are comparable but the convergence is slightly better with the agglomerated multigrid strategy. This is a bit surprising since the original multigrid algorithm employs a piecewise linear prolongation operator. A possible explanation is that the agglomeration algorithm creates better coarse grids than those employed in the non-nested algorithm. The CPU times required on the Cray Y-MP are 59 and 58 seconds with the original and the agglomerated multigrid, respectively, using three grids.

Perhaps the biggest advantage of the agglomeration algorithm lies in its ability to generate very coarse grids without any user intervention. Such extremely coarse grids should be beneficial in multigrid. Figure 10 shows two coarser grids for the four element airfoil case. These grids contain 63 and 22 vertices, respectively. With these grids it is now possible to use a 6 level agglomeration multigrid strategy. However, because these coarse grids are rather nonuniform, it is imperative that the first order coarse grid operator be a strictly positive scheme (i.e. one can no longer rely on assumptions of grid smoothness as conditions for stability). With the original first order operator in place, which is composed of a central difference plus a dissipative flux, it is difficult to guarantee the positivity of the scheme for arbitrary grids. In fact, the scheme has been found to be unstable on some of the very coarse and distorted agglomerated meshes. However, if the flux is replaced by a truly first order upwind flux, given for example by Roe's flux difference splitting [10], a stable scheme can be recovered for these coarse agglomerated grids. Thus, for each of the coarse grids obtained by agglomeration, a check of the convergence properties of the coarse grid operator at the desired flow conditions is carried out if problems are experienced with the multigrid. This step ensures that the coarse grid operators are convergent and that the problems with the multigrid, if any, come from the inter-grid communication. Figure 11 shows the convergence history with the 6 grid level agglomerated multigrid scheme. Also shown is the convergence with the 3 grid agglomeration multigrid scheme. In this particular case, Roe's upwind flux is

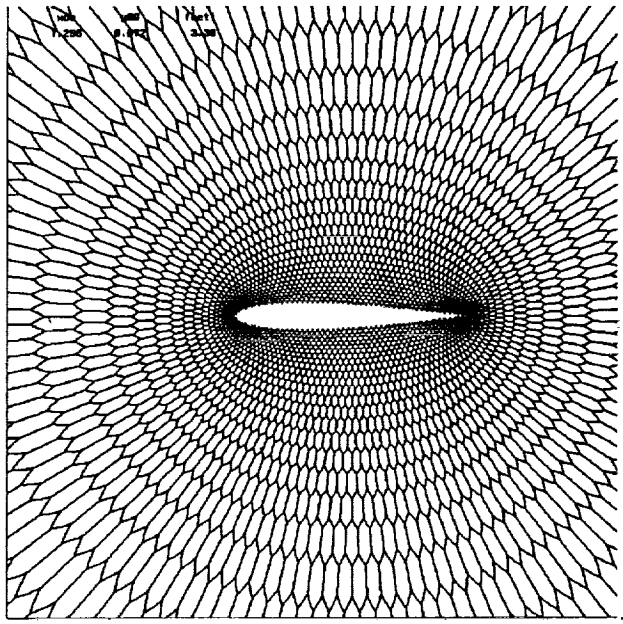

Figure 5: Dual to the fine grid having 4420 vertices.

used on the two coarsest grids, where central differencing proved unreliable. The time taken for the 6 grid agglomeration multigrid is 86 seconds. Thus the improved convergence rate is not entirely reflected in terms of the required computational resources. This is attributed to the increased time required by the Roe's upwind scheme, which involves a substantial number of floating point operations. This case serves to demonstrate the importance of the stability of each of the individual coarse grid operators in the multigrid scheme. Although first order upwinding has been employed on the distorted coarse meshes for demonstration purposes, it should be possible to construct stable central difference operators on such meshes.

5 Conclusions

It has been shown that the agglomeration multigrid strategy can be made to approximate the efficiency of the unstructured multigrid algorithm using independent, non-nested coarse meshes, in terms of both convergence rates and CPU times. It is further shown that arbitrarily coarse grids can be obtained with the agglomeration technique, although care must be taken to ensure that the coarse grid operator is convergent on these grids. Agglomeration has direct applications to three dimensions, where it may be difficult to derive coarse grids that conform to the geometry. In future work, alternate methods of generating coarse grids will be investigated. These may include the creation of maximal independent sets to create the coarse grid seed points and using these seed points to agglomerate the fine grid cells around them. A maximal independent set is a subset of the graph containing only vertices that are distance 2 apart in the original graph. Since coarsening algorithms can be viewed as partitioning strategies, there also exists a possible interplay between agglomerated multigrid techniques and distributed memory parallel implementations of the algorithm, which should be further investigated. Finally, the implementation of the viscous terms for Navier-Stokes flows on arbitrary polygonal control volumes must be carried out for this type of strategy to be applicable to viscous flows.

6 Acknowledgements

The first author thanks the NAS Applied Research Branch at NASA Ames Research Center for funding this project under contract NAS 2-12961. The second author's work was supported under NASA contract NAS1-19480. The third author was partially supported by AFOSR 91-0063, by DOE grant DE-FG02-88ER25053, by a PYI, NSF ASC-8858101 and by the Research Institute for Advanced Computer Science (RIACS) under Cooperative Agreement NCC2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA).

References

[1] R.H. Ni. A Multiple Grid Scheme for Solving the Euler Equations Proc *AIAA Journal, Vol 20,* pp. 1565-1571, 1982.

[2] A. Jameson. Solution of the Euler Equations by a Multigrid Method. In *Applied Math. and Comp.*, Vol. 13, pp. 327–356, 1983.

[3] D. J. Mavriplis and A. Jameson. Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes. In *AIAA Journal, Vol 26, No. 7,* July 1988, pp. 824-831.

[4] D. J. Mavriplis. Three Dimensional Multigrid for the Euler Equations. Proc *AIAA 10th Comp. Fluid Dynamics Conf.*, Honolulu, Hawaii, June 1991, pp. 239–248.

[5] M. P. Leclercq. Resolution des Equations d'Euler par des Methodes Multigrilles Conditions aux Limites en Regime Hypersonique. *Ph.D Thesis, Applied Math, Universite de Saint-Etienne*, April, 1990.

[6] J. Peraire, J. Peiro, and K. Morgan. A 3D Finite-Element Multigrid Solver for the Euler Equations. *AIAA paper 92-0449,* January 1992.

[7] M. Lallemand, H. Steve, and A. Dervieux. Unstructured Multigridding by Volume Agglomeration: Current Status. In *Computers and Fluids*, Vol. 21, No. 3, pp. 397–433, 1992.

[8] W. A. Smith. Multigrid Solution of Transonic Flow on Unstructured Grids. Proc *Recent Advances and Applications in Computational Fluid Dynamics, Proceedings of the ASME Winter Annual Meeting, Ed. O. Baysal,* November, 1990.

[9] G. Spragle, W. A. Smith, and Y. Yadlin. Application of an Unstructured Flow Solver to Planes, Trains and Automobiles. *AIAA Paper 93-0889,* January 1993.

[10] P. L. Roe. Approximate Reimann Solvers, Parameter Vectors, and Difference Schemes. In *Journal of Computational Physics, Vol. 43,* pp. 357-372, 1981.

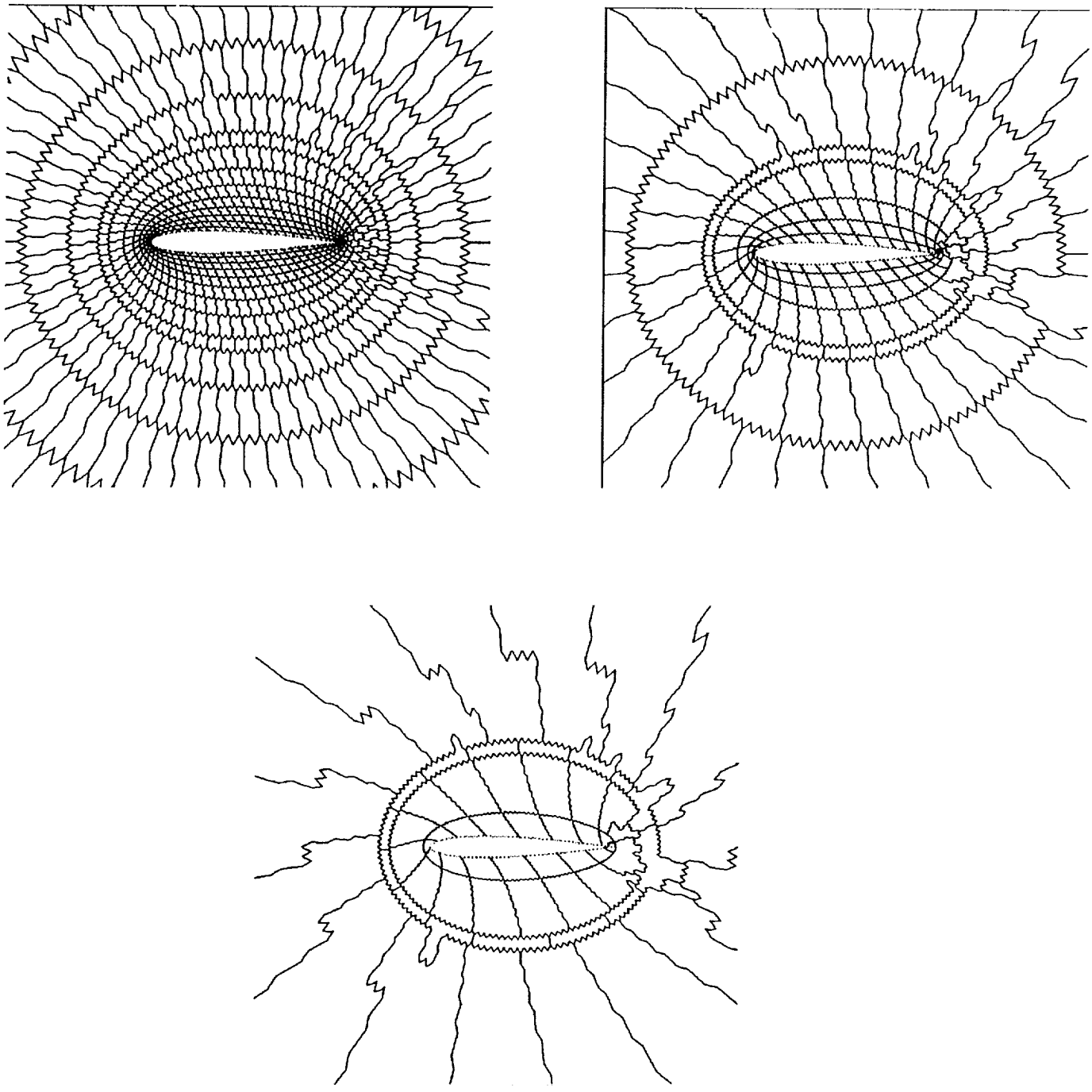

Figure 6: Three agglomerated coarse grids for the NACA0012 test case.

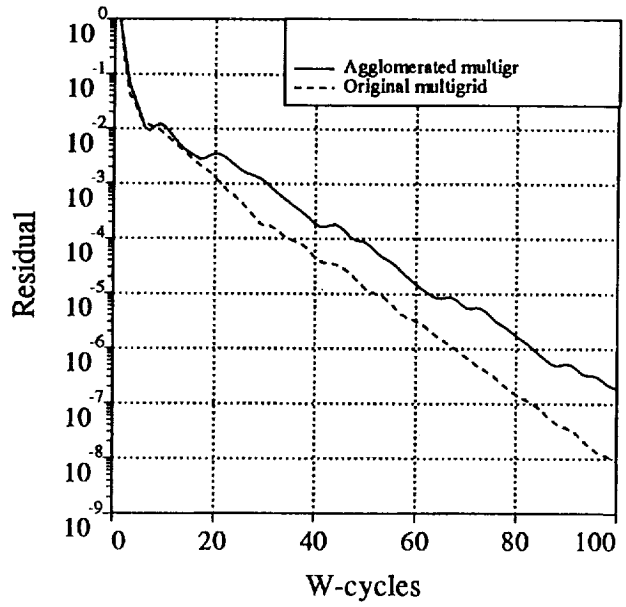

Figure 7: Convergence histories with the agglomerated and original multigrid.

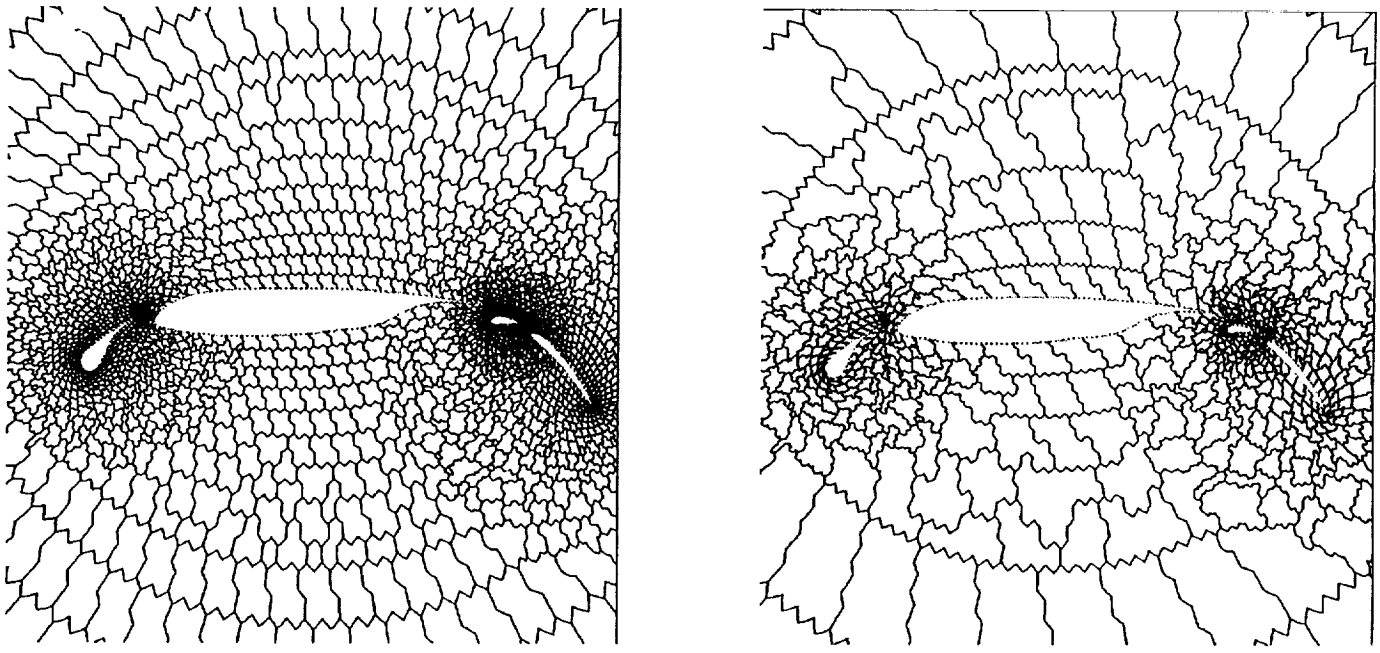

Figure 8: Two agglomerated coarse grids for the four-element test case.

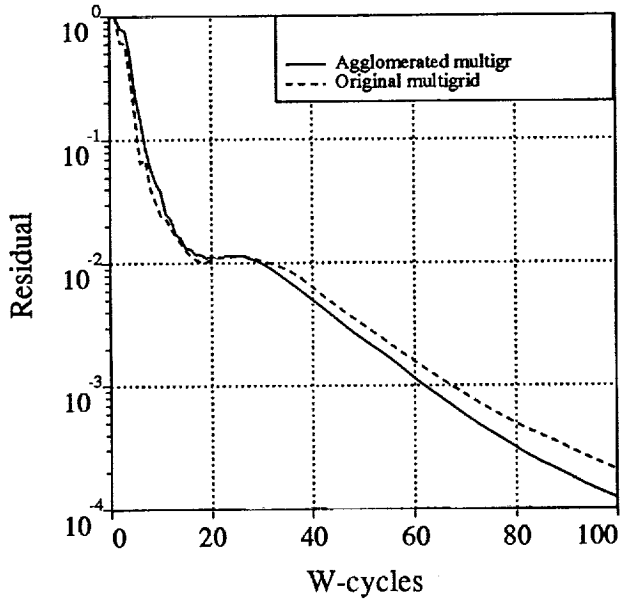

Figure 9: Convergence histories with the agglomerated and original multigrid.

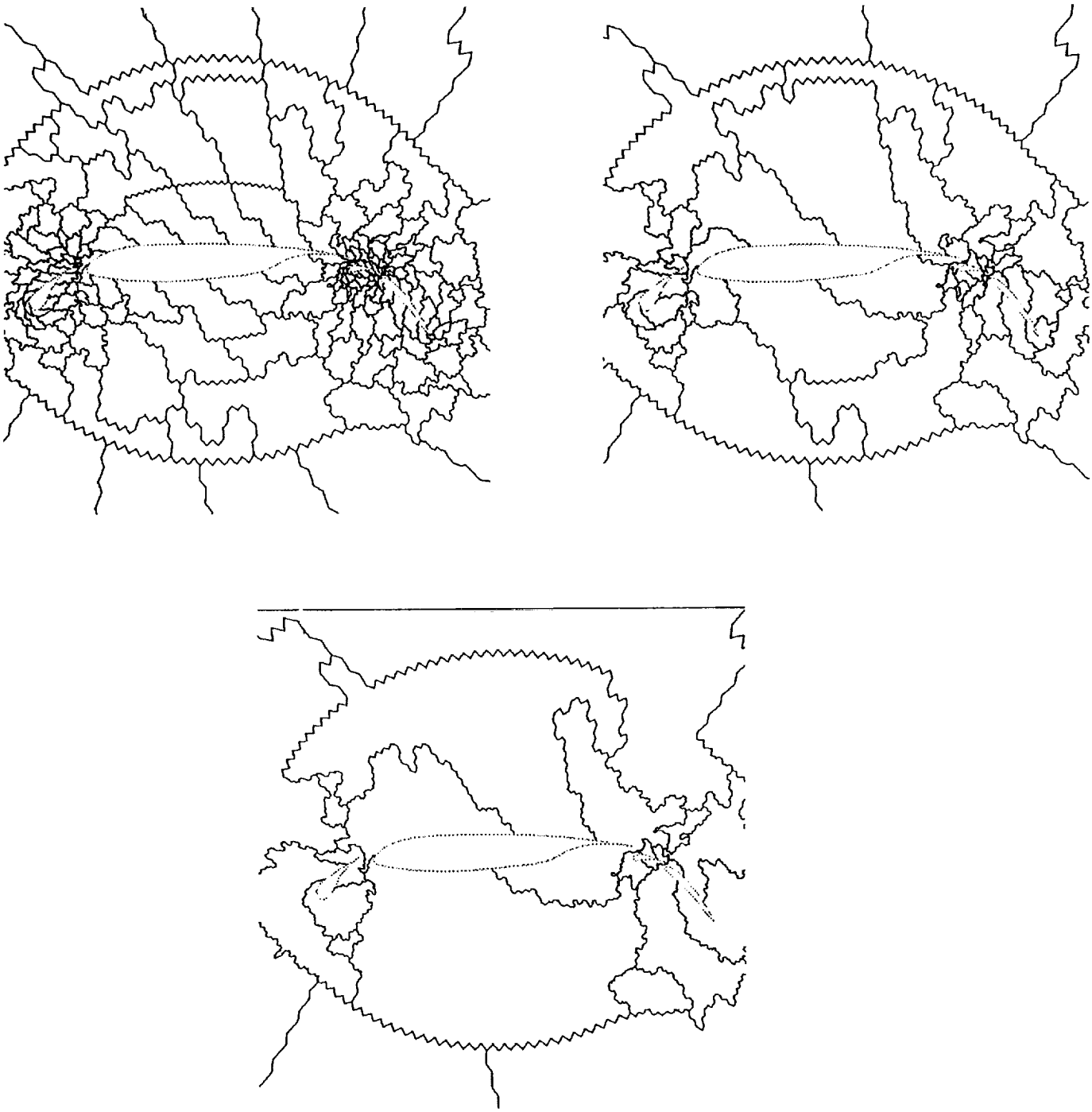

Figure 10: Three coarser grids for the four-element test case.

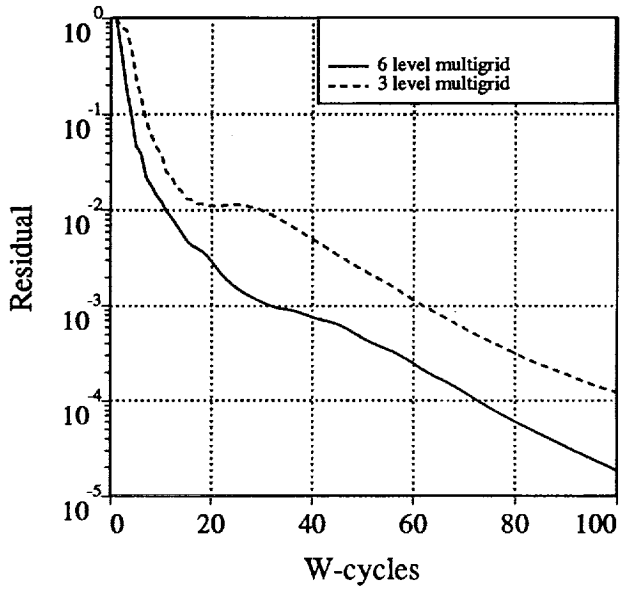

Figure 11: Convergence histories with the 6-level and 3-level agglomerated multigrid algorithms.